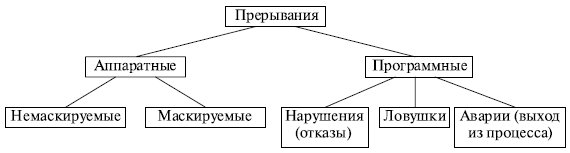
**Лекция 6. Прерывания и особые случаи**

**Прерывание** - это изменение естественного порядка выполнения программы, которое связано с необходимостью реакции системы на работу внешних устройств, а также на ошибки и особые ситуации, возникшие при выполнении программы. При этом вызывается специальная программа - **обработчик прерываний**, специфическая для каждой возникшей ситуации, после выполнения которой возобновляется работа прерванной программы.

Механизм прерывания обеспечивается соответствующими аппаратно-программными средствами компьютера.

Классификация прерываний представлена на [рис. 7.1](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=1#image.7.1).



**Рис. 7.1.** Классификация прерываний

Запросы аппаратных прерываний возникают асинхронно по отношению к работе микропроцессора и связаны с работой внешних устройств.

Запрос от **немаскируемых прерываний** поступает на вход NMI микропроцессора и не может быть программно заблокирован. Обычно этот вход используется для запросов прерываний от схем контроля питания или неустранимых ошибок ввода/вывода.

Для запросов **маскируемых прерываний** используется вход INT микропроцессора. Обработка запроса прерывания по данному входу может быть заблокирована сбросом бита IF в **регистре флагов** микропроцессора.

**Программные прерывания**, строго говоря, называются исключениями или особыми случаями. Они связаны с особыми ситуациями, возникающими при выполнении программы (отсутствие страницы в оперативной памяти, нарушение защиты, переполнение), то есть с теми ситуациями, которые программист предвидеть не может, либо с наличием в программе специальной команды INT n, которая используется программистом для вызова функций операционной системы либо BIOS, поддерживающих работу с внешними устройствами. В дальнейшем при обсуждении работы системы прерываний мы будем употреблять единый термин "прерывание" для аппаратных прерываний и исключений, если это не оговорено особо.

Программные прерывания делятся на следующие типы.

**Нарушение (отказ)** - особый случай, который микропроцессор может обнаружить до возникновения фактической ошибки (например, отсутствие страницы в оперативной памяти); после обработки нарушения программа выполняется с рестарта команды, приведшей к нарушению.

**Ловушка** - особый случай, который обнаруживается после окончания выполнения команды (например, наличие в программе команды INT n или установленный флаг TF в **регистре флагов** ). После обработки этого прерывания выполнение программы продолжается со следующей команды.

**Авария** (выход из процесса) - столь серьезная ошибка, что некоторый контекст программы теряется и ее продолжение невозможно. Причину аварии установить нельзя, поэтому программа снимается с обработки. К авариям относятся аппаратные ошибки, а также несовместимые или недопустимые значения в системных таблицах.

**Порядок обработки прерываний**

Прерывания и особые случаи распознаются на границах команд, и программист может не заботиться о состоянии внутренних рабочих регистров и устройств конвейера.

Реагируя на запросы прерываний, микропроцессор должен идентифицировать его источник, сохранить минимальный контекст текущей программы и переключиться на специальную программу - обработчик прерывания. После обслуживания прерывания МП возвращается к прерванной программе, и она должна возобновиться так, как будто прерывания не было.

Обработка запросов прерываний состоит из:

* "рефлекторных" действий процессора, которые одинаковы для всех прерываний и особых случаев и которыми программист управлять не может;
* выполнения созданного программистом обработчика.

Для того чтобы микропроцессор мог идентифицировать источник прерывания и найти обработчик, соответствующий полученному запросу, каждому запросу прерывания присвоен свой номер ( **тип прерывания** ).

**Тип прерывания** для программных прерываний вводится изнутри микропроцессора; например, прерывание по отсутствию страницы в памяти имеет тип 14. Для прерываний, вызываемых командой INT n, тип содержится в самой команде. Для маскируемых аппаратных прерываний тип вводится из **контроллера приоритетных прерываний** по шине данных. **Немаскируемому прерыванию** назначен тип 2.

Всего микропроцессор различает 256 **типов прерываний**. Таким образом, все они могут быть закодированы в 1 байте.

"Рефлекторные" действия микропроцессора по обработке запроса прерывания выполняются аппаратными средствами МП и включают в себя:

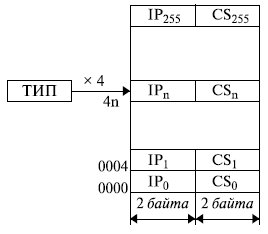
* определение **типа прерывания** ;
* сохранение контекста прерываемой программы (некоторой информации, которая позволит вернуться к прерванной программе и продолжить ее выполнение). Всегда автоматически сохраняются как минимум регистры EIP и CS, определяющие точку возврата в прерванную программу, и регистр флагов EFLAGS. Если вызов обработчика прерывания проводится с использованием шлюза задачи, то в памяти полностью сохраняется сегмент состояния TSS прерываемой задачи;
* определение адреса **обработчика прерывания** и передача управления первой команде этого обработчика.

После этого выполняется программа - **обработчик прерывания**, соответствующая поступившему запросу. Эта программа пишется и размещается в памяти прикладным или системным программистом. Обработчик прерывания должен завершаться командой I RET, по которой автоматически происходит переход к продолжению выполнения прерванной программы с восстановлением ее контекста.

Для вызова обработчика прерывания микропроцессор при работе в **реальном режиме** использует **таблицу векторов прерываний**, а в **защищенном режиме - таблицу дескрипторов прерываний**.

**Таблица векторов прерываний** ([рис. 7.2](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=1#image.7.2)) располагается в самых младших адресах оперативной памяти, имеет объем 1 Кбайт и содержит 4байтные элементы ( **векторы прерываний** ) для 256 обработчиков прерываний. Старшие 2 байта вектора загружаются в сегментный регистр команд CS, а младшие 2 байта - в **регистр указателя команд** IP. Обращение к элементам таблицы осуществляется по 8-разрядному коду - **типу прерывания**. Так как таблица всегда имеет нулевой начальный адрес и длину вектора в 4 байта, чтобы определить адрес вектора для прерывания типа i, достаточно просто умножить это значение на 4.

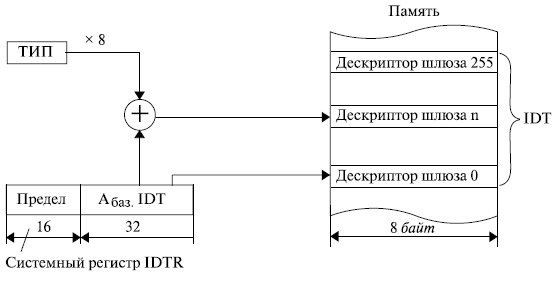
В **защищенном режиме** для вызова обработчика прерывания используется **таблица дескрипторов прерываний** IDT. Элементами таблицы являются 8-байтные дескрипторы типа шлюз – специальные программные структуры, через которые происходит передача управления обработчику.



**Рис. 7.2.** Таблица векторов прерываний

Обращение к IDT аналогично обращению к **глобальной таблице дескрипторов**, где вместо системного регистра GDT R используется регистр IDTR, который определяет размер и базовый адрес таблицы в памяти.

Физический адрес дескриптора шлюза, находящегося в IDT, определяется как сумма базового адреса таблицы и умноженного на 8 типа прерывания ([рис. 7.3](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=1#image.7.3)).



**Рис. 7.3.** Порядок обращения к таблице дескрипторов прерываний

Содержимое регистра IDTr не сохраняется в сегментах TSS и не изменяется при переключении задачи. Программы не могут обратиться к IDT, так как единственный бит TI **индикатора таблицы** в **селекторе** сегмента обеспечивает выбор только между таблицами GDT и LDT.

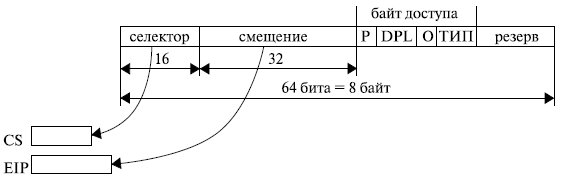
Максимальный предел таблицы дескрипторов прерываний составляет 256\*8 - 1 = 2047.

Можно определить предел меньшим, но это не рекомендуется. Если происходит обращение к дескриптору вне пределов IDT, процессор переходит в режим отключения до получения сигнала по входу NMI или сброса.

В IDT могут храниться только дескрипторы следующих типов:

* шлюз ловушки,
* шлюз прерывания, шлюз задачи.

Шлюзы ловушки и прерывания сходны со **шлюзом вызова**, только в них отсутствует поле счетчика WC ([рис. 7.4](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=1#image.7.4)). Так как прерывание является неожиданным событием и не связано с текущей программой, говорить о передаче параметров их обработчику не приходится.



**Рис. 7.4.** Формат шлюзов ловушки и прерывания

Бит S = 0 в байте доступа определяет этот дескриптор как системный объект. Если поле ТИП в **байте доступа** равно 1110, то это **шлюз прерывания**, если 1111 - то **шлюз ловушки**.

Поле **уровня привилегий** дескриптора DPL, как правило, устанавливается равным 3 с тем, чтобы к **обработчику прерываний** могли обращаться программы с любого уровня привилегий.

**Бит присутствия** P может быть равен как 0, так и 1.

При входе в обработчик через шлюз прерывания в регистре флагов сбрасывается **бит разрешения прерываний** IF. В этом случае микропроцессор блокирует все **маскируемые аппаратные прерывания**. Поэтому в обработчике прерываний этот бит должен быть установлен в 1 как можно раньше с тем, чтобы не блокировать работу программ, которые вызываются, например, при обработке прерываний от системного таймера.

При входе в обработчик через шлюз ловушки флаг IF не меняется.

Вызов обработчика через шлюз ловушки, а не шлюз прерывания, чаще реализуют при обработке исключений, так как на период обслуживания прерывания нежелательно выключать механизм разделения времени, использующий прерывания таймера.

Вызов обработчика через **шлюз задачи** обычно осуществляется при обработке аппаратных прерываний, так как такая обработка не связана с текущей выполняемой задачей. При этом возможен механизм вложенных прерываний, если прерывания в задаче разрешены. Вызов обработчика прерывания через шлюз задачи осуществляется и при обработке исключений, например, "неразрешенный TSS ", когда поврежденная задача не может вызвать процедуру прерывания. Переключение задач требует примерно в 5 раз больше времени, чем вызов процедуры. Поэтому, если приоритет запроса высок, а программа обслуживания короткая, ее оформляют в виде процедуры.

**Контроллер приоритетных прерываний**

Прерывание - один из наиболее дефицитных ресурсов в микропроцессорной системе. Микропроцессор имеет только 2 входа для приема запросов прерываний: вход INT - по нему принимаются запросы, обработка которых может быть замаскирована сбросом флага IF в регистре флагов, - и вход немаскируемых прерываний NMI. Вход NMI фактически закреплен за запросами прерываний от схем контроля питания. Поэтому при такой архитектуре микропроцессора в микропроцессорной системе обязательно должны использоваться средства, которые позволяют предварительно обрабатывать и передавать на вход маскируемых прерываний INT микропроцессора запросы от многочисленных внешних устройств, входящих в состав микропроцессорной системы. В качестве такой схемы используется **контроллер приоритетных прерываний** (КПП).

Мы рассмотрим его функционирование на примере БИС i8259, которая, с одной стороны, имеет самостоятельное значение, а с другой стороны, фактически без изменений входит в состав современных чипсетов.

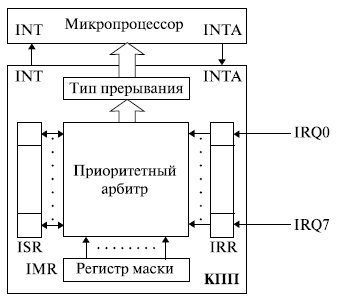
Его структура представлена на [рис. 7.5](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=2#image.7.5).

Функции контроллера приоритетных прерываний:

* восприятие и фиксация до 8 запросов прерываний (IRQ0 - IRQ7), поступающих по внешним входам;
* выделение наиболее приоритетного из поступивших запросов, включая возможность маскирования отдельных запросов;
* выдача на шину данных (по требованию микропроцессора) **типа** выбранного **прерывания**.

При использовании КПП обработка запросов немаскируемых прерывание проходит следующие этапы:

1. Системная периферия на системной плате или устройство ввода/вывода на внешней шине активирует одну из линий IRQx.

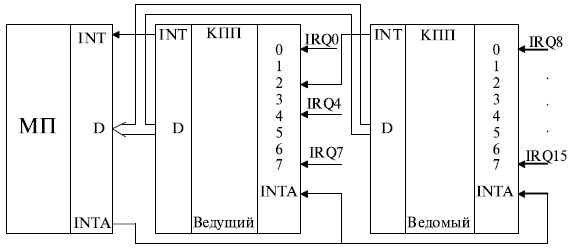


**Рис. 7.5.** Структура контроллера приоритетных прерываний

1. В **регистре запросов прерываний** IRR, который предварительно настраивается на восприятие запросов по спаду или низкому уровню сигнала, происходит установка соответствующих разрядов в "1".
2. Незамаскированные в **регистре маски** IMR запросы передаются в приоритетный арбитр, замаскированные блокируются.
3. В соответствии с выбранной в процессе инициализации дисциплиной обслуживания **приоритетный арбитр** выделяет наиболее приоритетный запрос. При системном сбросе контроллера самый высокий приоритет устанавливается для запроса, приходящего по входу IRQ0, а самый низкий - по входу IRQ7.
4. Приоритет выделенного запроса сравнивается с приоритетом запроса, который в данный момент может обрабатываться микропроцессором (его номер установлен в **регистре обслуживания прерываний** ISR ). Если приоритет нового запроса выше либо в данный момент обслуживаемых запросов нет, то контроллер формирует сигнал прерывания INT в микропроцессор, в противном случае обработка запроса откладывается. В **регистре типа прерывания** формируется тип принятого к обработке запроса прерывания.
5. МП воспринимает запрос прерывания, и если флаг IF = 1, то по завершении текущей команды выполняет 2 цикла подтверждения прерывания, выдавая сигналы на выход INTA:
   * в 1-м цикле запрещается запись в IRR. В ISR устанавливается разряд, соответствующий принятому к обработке запросу, и сбрасывается разряд в IRR ;
   * во 2-м цикле **тип прерывания** передается в МП по разрядам D0D7 шины данных. Разрешается запись в IRR ;
6. МП принимает **тип прерывания** и использует его в качестве индекса при обращении к соответствующей **таблице прерываний** ( **таблице векторов** или **таблице дескрипторов прерываний** в зависимости от режима работы МП).
7. В соответствии с установленным в микропроцессоре режимом работы (реальном или защищенном) и механизмом вызова программы - обработчика прерывания МП сохраняет необходимую информацию о прерываемой программе и переходит к выполнению **обработчика прерывания**.
8. Команда IRET, завершающая **обработчик прерываний**, восстанавливает прежнее состояние микропроцессора и передает управление прерванной программе.

**Каскадное включение контроллеров приоритетных прерываний**

Для расширения количества запросов прерываний, которые могут быть подключены к микропроцессору, в микропроцессорной системе может быть использовано несколько КПП. Схема каскадного подключения двух контроллеров представлена на [рис. 7.6](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=2#image.7.6).



**Рис. 7.6.** Каскадное подключение контроллеров приоритетных прерываний к микропроцессору

К входу INT микропроцессора подключается выход INT ведущего контроллера. Выход INT ведомого контроллера подключается к одному из входов IRQi ведущего КПП на правах других запросов прерываний, поступающих на этот контроллер. В персональной ЭВМ всегда используются два контроллера приоритетных прерываний, причем ведомый КПП подключен к входу IRQ2 ведущего.

На [рис. 7.5](http://www.intuit.ru/studies/courses/604/460/print_lecture/10333) было дано схематическое представление контроллера приоритетных прерываний. Для того чтобы лучше понять функционирование **контроллеров приоритетных прерываний** в реальных микропроцессорных системах и оценить все имеющиеся у них возможности, рассмотрим структуру КПП более подробно.

Регистры КПП делятся на 2 группы: регистры инициализации ICW1-ICW4 и операционные регистры OCW1-OCW3.

**Регистры инициализации** загружаются при инициализации контроллера и в процессе работы КПП не меняются.

Регистр ICW1 - управление микросхемой:

* настраивает контроллер на восприятие сигналов запроса по низкому уровню или заднему фронту;
* определяет, используется в МПС единственный КПП или применяется их каскадное включение;
* определяет порядок загрузки приказов инициализации. Необходимость этого обусловлена тем, что в пространстве ввода/вывода каждому контроллеру выделено всего 2 адреса. Так, в стандартной конфигурации персональной ЭВМ ведущему контроллеру выделены адреса 20h и 21h, а ведомому - A0h и A1h. В то же время каждый контроллер имеет в своем составе 7 регистров, к которым должен быть обеспечен программный доступ. В частности, при инициализации необходимо занести информацию в 4 регистра ICWi.

Регистр ICW2 - регистр **типа прерывания**. При инициализации в 5 его старших разрядов заносится некоторая константа. В процессе обработки запросов прерываний в 3 младшие разряда этого регистра заносится номер входа IRQi, по которому принят запрос, подлежащий обработке ([рис. 7.7](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=2#image.7.7)):

Формат регистра типа прерывания ICW2

**Рис. 7.7.** Формат регистра типа прерывания ICW2

В персональной ЭВМ в регистр ICW2 ведущего контроллера при инициализации заносится константа 00001b, а в ведомый - константа 01110b. Поэтому типы всех прерываний, запросы от которых поступают через ведущий КПП, лежат в диапазоне 00001000b-00001111b (08h-0Fh), а через ведомый - в диапазоне 01110000b-01110111b (70h-77h). Распределение входов прерываний в стандартной конфигурации персональной ЭВМ представлено в [табл. 7.1.](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=2#table.7.1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица 7.1. Распределение входов запросов прерываний в персональной ЭВМ | | | | |
| **Контроллер** | **IRQ** | **Тип** | **Назначение** | **Начальное состояние маски** |
| Ведущий контроллер | 0 | 8 | Таймер | 0 |
| 1 | 9 | Клавиатура | 0 |
| 2 | A | Ведомый КПП | 0 |
| 3 | B | Последовательный порт 2 (COM2) | 1 |
| 4 | C | Последовательный порт 1 (COM1) | 1 |
| 5 | D | Последовательный принтер (LPT2) | 1 |
| 6 | E | НГМД | 0 |
| 7 | F | Параллельный принтер (LPT1) | 1 |
| Ведомый контроллер | 8 | 70 | Часы реального времени | 1 |
| 9 | 71 | Резерв. Программно переназначен на IRQ2 | 0 |
| 10 | 72 | Резерв | 1 |
| 11 | 73 | Резерв | 1 |
| 12 | 74 | Мышь | 1 |
| 13 | 75 | Ошибка сопроцессора | 1 |
| 14 | 76 | НЖМД | 0 |
| 15 | 77 | Резерв | 1 |

Регистр ICW3 - регистр управления ведомым. Имеет различное назначение в ведущем и ведомом КПП. В ведущем КПП устанавливаются единицы в разрядах, соответствующих линиям с подключенными ведомыми КПП. В персональной ЭВМ его значение имеет вид 00000100b. В ведомом КПП пять старших разрядов этого регистра установлены в 0, а в трех младших кодируется номер входа ведущего КПП, к которому подключен данный ведомый. В персональной ЭВМ его значение имеет вид 00000010b.

Регистр ICW4 - регистр управления режимом. Определяет, является данный КПП ведущим или ведомым, тип окончания прерывания, то есть кем должен сбрасываться бит запроса в регистре обслуживания прерывания ISR, и другие параметры работы.

Так как для КПП определено только 2 допустимых состояния (ведущий или ведомый), максимальная конфигурация контроллеров приоритетных прерываний состоит из 1 ведущего и 8 ведомых КПП. Это обеспечивает возможность подключения к входу INT микропроцессора до 64 запросов прерываний.

Содержимое **операционных регистров** изменяется в процессе работы КПП записью в них новой информации.

Регистр OCW1 ( IMR ) - **регистр маски прерывания**. Код 1 в разряде i запрещает, а код 0 - разрешает обработку запроса прерывания по входу IRQi.

Регистр OCW2 - определяет один из трех возможных порядков изменения приоритетов запросов прерываний:

* приоритеты не меняются в процессе работы КПП;
* приоритеты меняются циклически: после обработки очередного запроса его приоритет становится самым низким, а приоритеты остальных запросов циклически сдвигаются;
* процессе работы КПП какому-либо запросу можно задать наивысший приоритет, приоритеты остальных запросов при этом циклически сдвигаются.

В персональной ЭВМ установлено постоянство приоритетов запросов прерываний, при этом запрос IRQ0 имеет самый высокий приоритет.

Так как здесь используются два каскадно включенных контроллера (см.[рис. 7.6](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=2#image.7.6)), приоритеты запросов прерываний IRQi имеют вид, представленный на [рис. 7.8](http://www.intuit.ru/studies/courses/604/460/lecture/10333?page=2#image.7.8).

Приоритетность запросов прерываний IRQi в персональной ЭВМ 

**Рис. 7.8.** Приоритетность запросов прерываний IRQi в персональной ЭВМ

Регистр OCW3 - управляет переводом контроллера в режим неприоритетного обслуживания и считыванием содержимого **регистра запросов** IRR и **регистра обслуживания** ISR. В режиме неприоритетного обслуживания микропроцессор получает от КПП только сигнал запроса прерывания, после чего микропроцессор должен программно считать содержимое регистров IRR и ISR и по своим алгоритмам определить, какой из имеющихся запросов прерываний принять к обслуживанию.

**Краткие итоги**. В лекции рассмотрена классификация прерываний, порядок обработки прерываний в реальном и защищенном режимах работы микропроцессора, структура и функционирование контроллера приоритетных прерываний.